



# EXPRESS MAIL LABEL NO: EL579667004US

METHOD AND SYSTEM FOR INSERTING A DATA OBJECT INTO A COMPUTER-GENERATED DOCUMENT USING A TEXT INSTRUCTION

5

15

20

25

30

Thomas Lange

### BACKGROUND OF THE INVENTION

## 10 Field of the Invention

The present invention relates generally to generating documents using a computer application, and in particular to inserting a data object like a mathematical formula or special characters like Greek characters into a computer-generated document as for example a text document.

#### Description of Related Art

Computer word processing applications typically are used to generate a document, referred to as a computer-generated document, that may contain text data, tables, diagrams, etc. and often mathematical formulae or special characters like Greek characters. Mathematical formulae and special characters are particularly important for documents like scientific articles and the like. Similarly, HTML Web page generators generate a document that is effectively a text-based document.

For creating a mathematical formula within a text document 100 (Fig. 1), so called formula editors were used. Typically, the formula editor was opened from within the computer word processing application by clicking on a menu bar icon, or alternatively using a menu.

10

15

20

25

30

35





The formula editor contained a large number of displayed key fields and list boxes representing different elements of mathematical formulae like brackets, integrals, fraction bars, matrices, so forth. For inserting special characters, like for example the Greek character  $\Sigma$ , it was necessary to enter a list box containing the special characters.

The user created the desired formula 101 using these keys and list boxes. After having completed the formula, the user returned to the original document and pasted the formula as an imported object into the document. If the user recognized an error in the formula, the user again opened the formula editor, corrected the error, and returned to the original document.

Using a formula editor, it was possible to create nearly every desired mathematical formula; however, the operation was complicated and time consuming in particular for simple formulae like simple fractions or square roots, which appeared frequently in a text document. Editing of the formula always required entering the formula editor and subsequently returning into the original document.

To simplify the entry of formulas, some formula editors permitted the use of script like phrases that the formula editor converted to the corresponding mathematical expression. However, while this assisted in entering a formula in some situations by minimizing the use of key fields and list boxes, the general problem of having to utilize the formula editor persisted.

In an attempt to minimize some of the entry and exit issues, it was known to select an insert option from a menu bar of an application and the formula editor capability was opened so that the user and

10

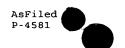
15

20

25

30

35





insert could edit a formula without leaving the application, only the menus and the object bars were changed. After the formula editor capability was used to enter the data object, double clicking on the embedded object launched the formula editor capability so that the formula could be edited. Again, this was done without leaving the application.

## SUMMARY OF THE INVENTION

According to the principles of this invention, inserting or editing a data object like a mathematical formula or special character in a computer-generated document is facilitated and sped up in comparison to the prior art methods that required use of a formula editor. A method of inserting a data object into a computer-generated document includes inputting instruction symbols representing the data object into the document in the form of text characters, selecting the document portion containing instruction symbols, and converting the instruction symbols contained in the selected document portion into a data object represented by the instruction symbols.

With the present invention it is possible to input the data object, which may be a mathematical formula or a Greek, Chinese, Korean, Cyrillic, Arabic, Hebrew, or Japanese character, or any other character or symbol, and which can be represented by certain instruction symbols, into the document using standard characters, which are also used for creating a text document. The user does not need to leave the document and can input the instruction symbols in the same way as the text characters, for example by typing on a keyboard.

If the selected document portion contains characters, which are not part of an instruction these characters remain unchanged during the converting

10

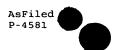
15

20

25

30

35





operation. Those unchanged characters may be variables like a, b, or x in a mathematical formula.

In one embodiment, the converted data object is inserted into the document at the position of the selected document portion. The inserted data object is formatted depending on a surrounding content, for example, the same as the format of text in the same line. The inserted data object is automatically stored with the document in this embodiment. The inserted data object is reconvertible into the original document portion for editing purposes.

The document portion including the instruction symbols may be input by means of speech decoding. In this case, the present invention is particularly advantageous since the instruction symbols (in contrast to the mathematical symbol itself) may be expressed orally.

One embodiment of the invention allows fast and easy generation and editing of a data object like a mathematical formula or special characters. This is particularly useful for simple and short data objects and for data objects, which the user needs frequently and for which the user easily memorizes the instruction symbols representing these data objects. For inserting the object, the user needs not to enter a special tool like a formula editor and then return to the original document. Another advantage of the present invention is that it allows the input of the data objects by speech decoding since the instruction symbols can be expressed orally.

Another embodiment of the invention provides a computer program for inserting, on a computer, a data object into a document, comprising inserting instruction symbols representing the data object in the form of text characters into the document, selecting a

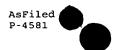
10

15

20

25

30





document portion containing instruction symbols, and converting the instruction symbols contained in the selected document portion into the data object represented by the instruction symbols.

Program code may be embodied in any form of a computer program product. A computer program product comprises a medium configured to store or transport computer readable code, or in which computer readable code may be embedded. Some example of computer program products are CD-ROM discs, ROM cards, floppy discs, magnetic tapes, computer hard drives, servers on a network and signals transmitted over a network representing computer readable program code.

According to a still further embodiment, the present invention provides a software tool providing instructions for inserting a data object into a computer-generated document by inserting instruction symbols inputted in the form of text characters and representing the data object into the document, converting instruction symbols contained in a selected document portion into the data object represented by the instruction symbols, inserting the converted data object into the document, and providing signals for displaying the document including the converted data object.

According to another embodiment, the present invention provides a computer-generated document including a data object generated by a conversion of instruction symbols inputted in the form of text characters, wherein the data object is reconvertible into the instruction symbols.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a schematic representation of a prior art document containing a mathematical formula.

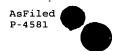




Figure 2A is an example of a text document containing instruction symbols representing a data object according to the present invention.

Figure 2B is a schematic representation of the text document shown in Figure 2A after conversion of a data object.

Figure 2C is a process flow diagram for the method of the present invention.

Figure 3A is a schematic illustration of a computer system to which the present invention may be applied.

Figure 3B is a schematic illustration of a clientserver computer system in which the present invention may be transferred and/or downloaded.

15

20

25

30

35

5

#### DETAILED DESCRIPTION

According to the principles of this invention, a user enters a formula in a computer-generated document by simply typing in text representing the formula and selecting this text. In response to the selection of the text representing the formula, the text representing the formula is automatically converted to a mathematical formula and inserted in the computer-generated document as a data object.

Consequently, with this invention, a user generating a document on a computer no longer has to continually open a formula editor to enter a formula. Rather, the user simply continues to input text information in the same form as the rest of the document including text that describes the formula. Similarly, a user can type in text representing a special character, e.g., a Greek, Chinese, Korean, Cyrillic, Arabic, Hebrew, or Japanese character, or any other character or symbol, and use the method of this invention to automatically convert the text

10

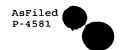
15

20

25

30

35





representing the special character to a data object that is inserted in the computer-generated document.

According to the principles of this invention, in a text-based formula generation method 205, a user inputs text in an input text operation 221 (Fig. 2C) into a computer-generated document 200A (Fig. 2A), which is displayed on a display screen 210 by an application 319 (Fig. 3A) executing on a computer processor 312C. In operation 221, (Fig. 2C) the user inputs the text using, for example, a keyboard in input units 320C (Fig. 3A) of a computer system 300C, which is representative of a computer system input device. The text, however, can be input using another suitable input technique and/or input device, e.g. voice recognition processing or the like.

Input text operation 221 transfers to formula check operation 222. If the user does not want to input a formula, formula check operation 222 returns to input text operation 221. Conversely, if the user wants to input a formula into document 200A, formula check operation 222, which is carried out by the user, transfers to input instruction operation 223.

In input instruction operation 223, the user inputs the formula using text instruction symbols via one of input units 320C. For example, as illustrated in Figure 2A, the user inputs the text portion "x equal sqrt a over b", which includes the text instruction symbols, equal, sqrt, and over. The user is not required to change modes of input, and is not required to access a formula editor and type the formula into the editor, but rather the user simply continues inputting characters in a conventional fashion.

After completing the text input for the desired formula in input instruction operation 223, the user selects the text formula instruction in select

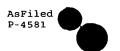
15

20

25

30

35





instruction 224. In this embodiment, the user first highlights text formula instruction 212 and then moves cursor 211 to an equation icon 213. With cursor 211 on equation icon 213 and with text formula instruction 212 highlighted, the user clicks a mouse button to complete select instruction operation 224. In more general terms, select instruction operation 224 identifies a text formula instruction 212 for a generate formula method 230. Operations 221 to 224 form a text formula instruction generation and identification method 220.

In generate formula method 230, formula check operation 231 determines whether the user selected a text formula instruction. In this embodiment, check operation 231 determines whether the user clicked on equation icon 213. If the user selected a text formula instruction, check operation 231 transfers to convert instruction operation 233 and otherwise to continue operation 232. In one embodiment, check operation 231 is part of an event handler of application 319, and if the event is not a text formula instruction selection input, event handling continues in continue operation 232 and the application continues as in the prior art.

However, if a text formula instruction selection input event occurred, processing transfers to convert instruction operation 233. Convert instruction operation 233 cuts the selected text formula instruction and pastes the selected text formula instruction into a call to a formula editor that can process the text formula instruction. For example, a prior art formula editor is modified to receive a text formula instruction and output a data object that is a corresponding formula. The modified formula editor executes in the background and the user is unaware of its existence. Upon the modified formula editor



returning a data object, which in this example is a mathematical formula

$$x = \sqrt{\frac{a}{b}} . ,$$

5

10

15

20

25

30

combinations of characters in the text formula instruction, which do not represent text instruction symbols, like the variables x, a and b in this example, remain unchanged. Hence, the creation of a formula containing variables is possible. Upon return of the mathematical formula, i.e., the data object, processing transfers from convert instruction operation 233 to insert formula operation 234.

In insert formula operation 234, the data object, i.e., formula 214, is inserted in document 200B at the location from which the text formula instruction sequence was cut, and is displayed on display unit 210. Preferably, the formula is formatted like the surrounding text so that the visual appearance of text document 200B containing the formula is optimized. However, in one embodiment, the user can include text instructions to format any part, or all of the formula in a specific format, which may be different from the format of the surrounding text.

Following insert formula operation 234, document complete check operation 235 determines whether the user has entered an instruction to indicate the document is complete. If a document complete instruction has been issued, the finished document is saved. Preferably, the inserted data object is stored together with the text document in a memory, e.g., memory 311B, which is this case is located in a file server 300B. If the document is not complete, check operation 235 returns to input text operation 221.

-9-





Those of skill in the art will appreciate that the method of this invention can be multithreaded. For example, one thread permits the user to continue entering additional text, while another thread executes the text formula instruction. Also, as illustrated in Figures 2A and 2B, the content of a text document 200A, may include in addition to the text data also other data like diagrams, graphics or tables. The text document also may be, for example, an HTML- or XML-document. In addition, the present invention is not restricted to text documents.

Hence, according to the principles of this invention, if a user wishes to input a special data object like a formula into the text document, the user enters the formula in the form of a text formula instruction that includes text instruction symbols and variables. For example, the formula

 $\frac{a}{b}$ 

20

25

30

10

15

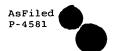
is represented by "a over b". Here, the characters "a" and "b" represent variables and "over" is a text instruction symbol representing a fraction bar. Other examples of text formula instructions are "sqrt a" for  $\sqrt{a}$ , "3 ind 1" for  $3_1$  and "int (a,b) Omega dt" for

$$\int_a^b \Omega dt$$
.

From the last example, it is obvious that the present invention is also very useful for inserting special characters like Greek characters into a text document. "pi" may represent the Greek character  $\pi$ , "alpha" may represent  $\alpha$  or "lambda" may represent  $\lambda$ . It is also

15

20





possible to distinguish between small and capital letters, "Lambda" may for example represent  $\Lambda$ . It is immediately apparent that typing the instruction symbols is in many cases much easier and faster than using a special program like a formula editor or a list box for Greek symbols. The same can apply to other special characters like Chinese, Korean, Cyrillic, Arabic, Hebrew, or Japanese characters, or any other character or symbol characters. In another embodiment, a character, e.g., a percent sign, is used before the name of the character to assist in distinguishing between when the user wants the text word, and when the user wants the Greek or other symbol.

Table 1 lists a number of different formula symbols that can be generated in using a text formula instruction. Notice that in each instance, the text formula instruction utilizes only characters that are found on a conventional computer keyboard. The last column in a row of Table 1 gives a simple example of a text instruction for a formula that utilizes the symbol presented in the first column of the row. In the last column, a, b, x, y, and z are used as variables. The text instruction symbol is in a bold font.

25 TABLE 1

Symbol			Example of
Presented	Troo	Description	text formula
in	Туре	Description	instruction
Formula			Instruction
	Unary	Dlug Gian	
+	operator	Plus Sign	+a
	Unary	Minua Cian	_
_	operator	Minus Sign	-a
±	Unary	Plus Minus Sign	<b>plusminus</b> a





		<del></del>	<del></del>
Symbol			Example of
Presented	Туре	Description	text formula
in	1,456	Description	instruction
Formula			Instruction
	operator		
<del>-</del>	Unary	Minus Dlus Gian	
<del>-</del>	operator	Minus Plus Sign	minusplus a
	Unary	Logical	
7	operator	negation	neg a
	Unary		
	operator/	Absolute value	<b>abs</b> a
·	function		
	Unary		
!	operator/	Factorial	<b>fact</b> a
·	function		
	Unary		
√	operator/	Square root	<b>sqrt</b> a
	function		_
			nroot n a
7 /	Unary ,		where n is the
n $\sqrt{}$	operator/	n-th root	desired nth
	function		root of a
	Unary	User-defined	
	operator	operator	uoper %theta x
	Binary		
=	operator/	Equal	a = b
	relation		
	Binary		_
<b>≠</b>	operator/	Not equal	a <b>neq</b> b, or
	relation		a <> b
	Binary		_
+	operator	Addition	a + b
	Binary	Add symbol in	
Φ	operator	circle	a <b>oplus</b> b
_	Binary	Substraction	a - b





Symbol			Example of
Presented	<b></b>	December	Example of
in	Type	Description	text formula
Formula			instruction
	operator		
	Binary	Subtract symbol	
θ	operator	in circle	a <b>ominus</b> b
	Binary	26 71 1 2	. 1
*	operator	Multiply	a * b
	Binary	2	
•	operator	Dot product	a <b>cdot</b> b
	Binary	Dot product in	
. 0	operator	a circle	a <b>odot</b> b
	Binary	Man 1 to 1 to 1 to 1 to 1	
×	operator	Multiplication	a <b>times</b> b
	Binary	Multiply symbol	a <b>otimes</b> b
⊗	operator	in circle	a otimes b
1	Binary	Division	a / h
/	operator	DIVISION	a / b
		Slash for	
,	Binary	quotient set	a <b>slash</b> b
/	operator	between two	slash c
		characters	
		Slash between	
		two characters,	
	Diw	of which the	
ه/ه	Binary	left character	a <b>wideslash</b> b
	operator	is superscript,	
		and the right	
		is subsript	
		Back Slash	
	D.t.	between two	
اره	Binary	characters, of	a <b>widebslash</b> b
,	operator	which the right	
		character is	



		<u></u>	1
Symbol Presented in Formula	Type	Description	Example of text formula instruction
		superscript,	
		and the left	
		subscript	
	Binary	Subscript	
Ø	-	Slash in circle	a <b>odivide</b> b
	operator		
÷	Binary	Division	a <b>div</b> b
	operator		
а Б	Binary	Division/	a <b>over</b> b
	operator	Fraction	
	Binary	Logical AND	a <b>and</b> b, or
	operator		'a&b
	Binary	Logical Or	a <b>or</b> b, or
, v	operator	Hogical of	a   b
0	Binary	Concetonate	a <b>circ</b> b
O	operator	Concatenate	a cire b
	Binary	<b>.</b>	- 1' 'I oo
	operator	Divides	5 <b>divides</b> 30
	Binary		
ł	operator	Does not Divide	7 <b>ndivides</b> 30
	Binary		
>	operator /	Greater than	a <b>gt</b> b, or
	Relation		a > b
	Binary		
<	operator /	Less than	a <b>le</b> b, or
	Relation		a < b
	Binary		
≥	operator /	Greater than or	a <b>gt</b> b, or
_	Relation	equal to	a >= b
	Binary	Greater than-	
≽	operator /	equal to	a <b>gtslant</b> b
	Relation		



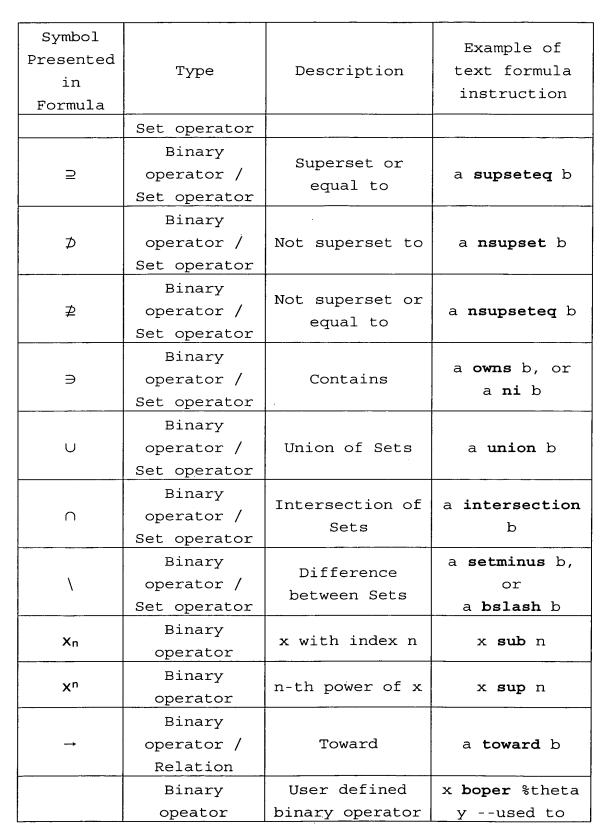


		<del></del>	<u> </u>
Symbol			Evample of
Presented	TT+	Doggariation	Example of
in	Туре	Description	text formula
Formula			instruction
	Binary		
<b>≫</b>	operator /	Much greater	a <b>gg</b> b, or
	Relation	than	a >> b
-	Binary		· · · · · · · · · · · · · · · · · · ·
≤	operator /	Less than or	a <b>le</b> b, or
_	Relation	equal to	a <= b
-	Binary	-	
ا ر ا	operator /	Less than-equal	a <b>leslant</b> b
≼	Relation	to	a restant n
		•	
	Binary	Morale loss than	a <b>11</b> b, or
≪	operator /	Much less than	a << b
	Relation	- 1 5	
	Binary	Is defined as/	<u>-</u>
def —	operator /	by definition	a <b>def</b> b
	Relation	equal to	
	Binary	Is equivalent/	
=	operator /	congruent to	a <b>equiv</b> b
	Relation	congruence co	
	Binary	Is	
≈	operator /		a <b>approx</b> b
	Relation	approximately	
	Binary		
~	operator /	Is similar to	a <b>sim</b> b
	Relation		
	Binary		
~	operator /	Is similar or	a <b>simeq</b> b
	Relation	equal to	- <b>4</b>
	Binary		
∞c	operator /	Is proportional	a <b>prop</b> b
	Relation	to	~ <b>Prop</b> D
		In orthogonal	a ortho h
<u> </u>	Binary	Is orthogonal	a <b>ortho</b> b





		· · ·	
Symbol Presented in Formula	Туре	Description	Example of text formula instruction
	operator /	to	
	Relation		
	Binary		·
	operator /	Is parallel to	a <b>parallel</b> b
	Relation		
	Binary		
•••	operator /	Correspondence	a <b>transl</b> b
	Relation	symbol image of	
	Binary	Correspondence	
0-0	operator /	symbol original	a <b>transr</b> b
	Relation	of	
	Binary		
€	operator /	Is contained in	a <b>in</b> b
	Set operator		
	Binary		
∉	operator /	Is not	a <b>notin</b> b
,	Set operator	contained in	
	Binary		
	operator /	Subset	a <b>subset</b> b
	Set operator		
·	Binary		
⊆	operator /	Subset or equal	a <b>subseteq</b> b
	Set operator	to	•
	Binary		
⊄	operator /	Not subset to	a <b>nsubset</b> b
;	Set operator		
	Binary		
⊈	operator /	Not subset or	a <b>nsubseteq</b> b
	Set operator	equal to	*
	Binary		
) >	operator /	Superset	a <b>supset</b> b
L	· , - '.		







_ <del></del>		<del></del>	I
Symbol Presented in	Туре	Description	Example of text formula
Formula			instruction
ronmara			insert greek character theta
Σ	Operator	Sum	Sum x sub i
П	Operator	Product	prod x sub i
П	Operator	Coproduct	coprod x sub I
lim	Operator	Limit	lim x toward infinity
lim inf	Operator	Limit inferior	liminf
lim sup	Operator	Limit superior	limsup
1-	Operator/	Physics	
ħ	Physics	Constant	hbar
	Operator/	Physics	
λ	Physics	Constant	lambdabar
3	Operator/ Logic	Existential quantifier, there is at least one	a <b>exists</b> b
A	Operator/ Logic	Universal quantifier, for all	a <b>forall</b> b
<b>(</b> =	Operator/ Logic	Arrow with double line to the left	a <b>dlarrow</b> b
⇒	Operator/ Logic	Arrow with double line to the right	a <b>drarrow</b> b
⇔	Operator/ Logic	Arrow with double line to the left and to the right	a <b>dlrarrow</b> b





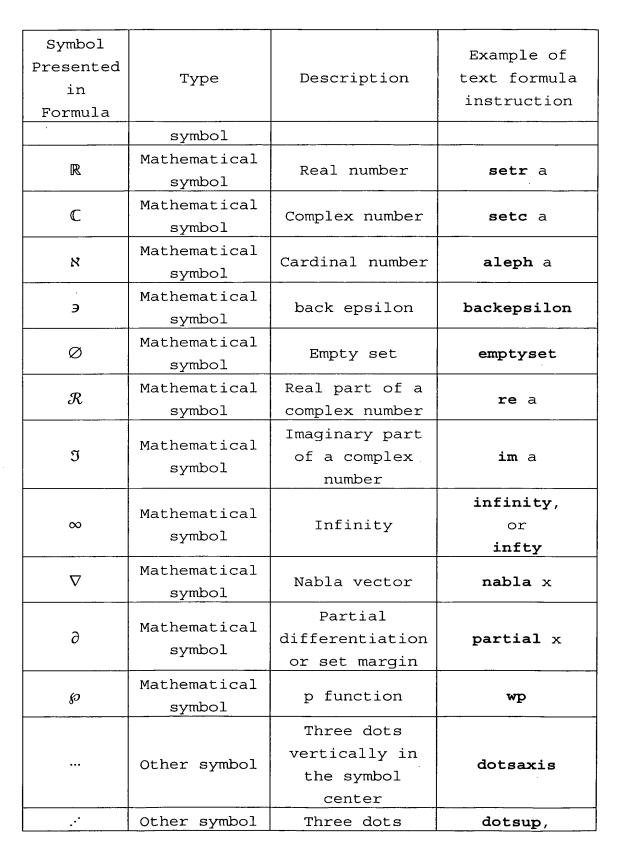
			T
Symbol Presented in Formula	Туре	Description	Example of text formula instruction
1	Operator	Up arrow	a <b>uparrow</b> b
ļ	Operator	Down arrow	a <b>downarrow</b> b
<b>←</b>	Operator	Left arrow	a leftarrow b
<b>→</b>	Operator	Right arrow	a rightarrow b
ſ	Operator	Integral	in xdx
IJ	Operator	Double Integral	iint f(x,y)dxdy
M	Operator	Triple Integral	<pre>iiint f(x,y,z)dxdydz</pre>
∮	Operator	Curve integral	lint
∯	Operator	Double curve	llint
∰	Operator	Triple curve integral	lllint
	Operator	User defined operator	<pre>oper %union from {i=1} to     n x_{i}</pre>
	Operator	Range from	from {i=1} to n
,	Operator	Lower limit of an operator	<b>from</b> {i=1}
	Operator	Upper limit of an operator	to n
sin()	Function	Sine	sin x
cos()	Function	Cosine	cos x
tan()	Function	Tangent	tan x
cot()	Function	Cotangent	cot x
arcsin()	Function	Arcsine	arcsin x
arccos()	Function	Arccosine	arccos x
arctan()	Function	Arctangent	arctan x

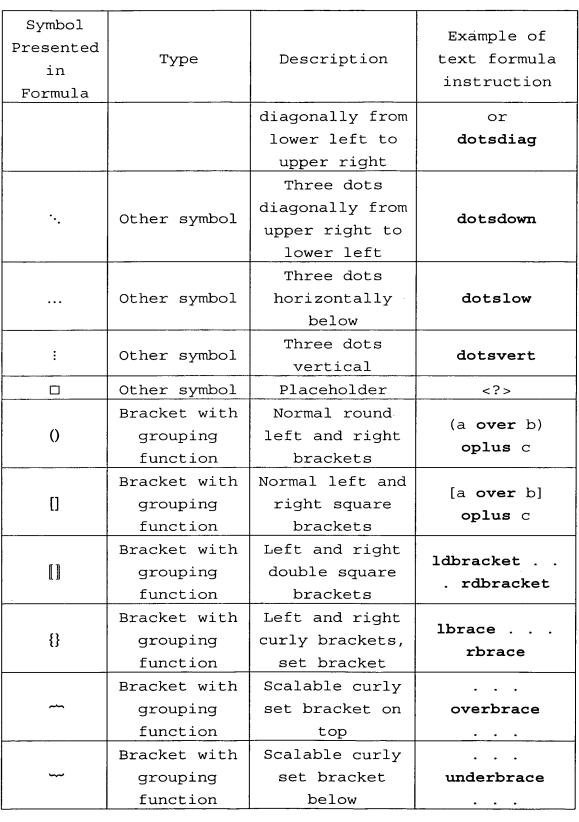




Symbol Presented in Formula  arccot()  sinh()  Function  Function  Type  Description  Description  Arccotangent  Arccotangent  Arccot x  Sinh()  Function  Hyperbolic sine  Sinh x	a
	-
sinh() Function Hyperbolic sine sinh x	
1	
cosh() Function Hyperbolic cosh x	
tanh() Function Hyperbolic tanh x	
coth() Function Hyperbolic coth x	
arsinh() Function Area hyperbolic sine arsinh x	
arcosh() Function Area hyperbolic cosine arcosh x	
artanh() Function Area hyperbolic tanget artanh x	
arcoth() Function Area hyperbolic cotangent arcoth x	
exp() Function exponential exp x function	
In Function Natural logarithm ln x	
log Function Logarithm log x	
e Function exponential func e^{x}	
IN Mathematical Natural number setn a symbol	
<pre>Mathematical</pre>	
Mathematical Rational number setq a	



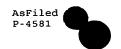


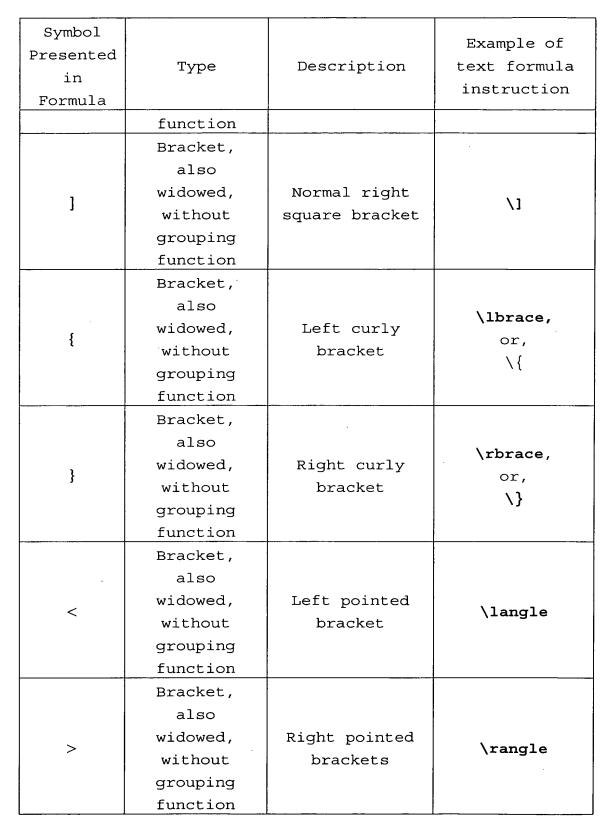




	<u> </u>		,
Symbol Presented in Formula	Type	Description	Example of text formula instruction
<>	Bracket with grouping function	Left and right pointed brackets	langle rangle
< >	Bracket with grouping function	Left and right pointed operator brackets	langle mline rangle
<   >	Bracket with grouping function	Scalable left and right pointed operator brackets	<pre>left langle mline right   rangle</pre>
	Bracket with grouping function	Left and right vertical lines	lline rline
	Bracket with grouping function	Left and right double lines	ldline rdline
Į J	Bracket with grouping function	Left and right lines with lower edges	lfloor rfloor
ΓŢ	Bracket with grouping function	Left and right lines with upper edges	lceil rceil
	Bracket with grouping function	Automatic sizing of brackets by putting left and right (left right) up front,	

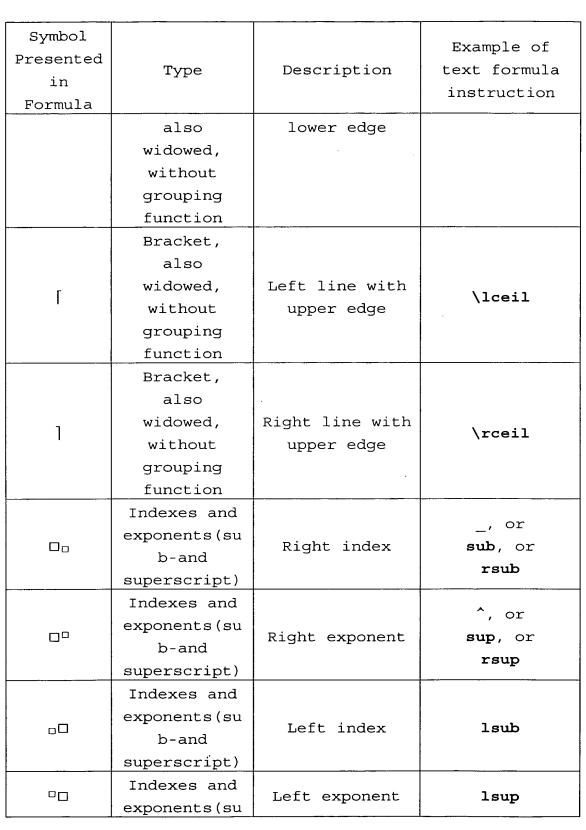
Comple - 1			
Symbol			Example of
Presented	Type	Description	text formula
in		_	instruction
Formula			
		e.g., left(a	
		over b right)	
		or left lceil .	
		right	
		lceil. This	
		way round,	
		square, double	
		square, single,	
		double, single,	
		curley,	
		pointed, and	
		operator	
		brackets can be	
		changed.	
	Bracket,		
	also		
,	widowed,	round left	
(	without	bracket	\(
	grouping		
	function	:	
	Bracket,		
	also		
	widowed,	Normal round	• -
)	without	right bracket	\)
	grouping	_	
	function		
	Bracket,	-	
	also	Normal left	
l r	widowed,		1/
•	without		\ L
	grouping		
	2+005+113		

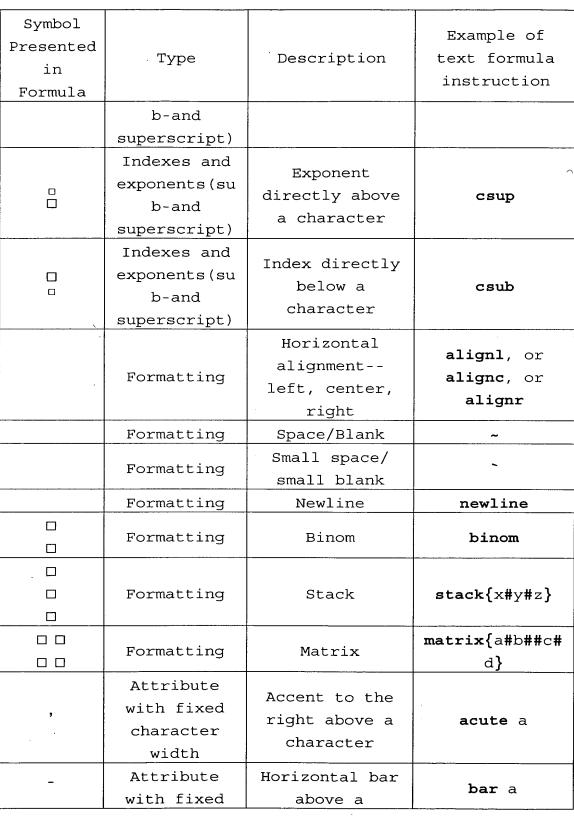






Symbol Presented in Formula  Bracket, also  Example of text formula instruction	ıla
Presented Type Description text formula instruction  Bracket,	ıla
in instruction instruction instruction in instructi	
Formula  Bracket,	
also	
widowed, Left vertical \lambdalline	
without   line   \land	
grouping	
function	
Bracket,	
also	
widowed, Right vertical	
without line \rline	
grouping	
function	
Bracket,	
also	
widowed, Left double	
grouping	
function	
Bracket,	
also	,
widowed, Right double	
without lines \rdline	
grouping	
function	
Bracket,	
also	
widowed, Left line with	
\lambda	
grouping	
function	
J Bracket, Right line with \rfloor	







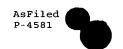


		<u> </u>	
Symbol			Example of
Presented	Туре	Description	text formula
in			
Formula			instruction
	character	character	
	width		
	Attribute		<b>breve</b> a
	with fixed	Upside down	
	character	roof above a character	
	width		
	Attribute		
,	with fixed	Upside down	
, i	character	roof	check
	width		
	Attribute		
	with fixed	Circle above a	circle a
•	character	character	
	width		
	Attribute		
	with fixed	Dot above a	
• .	character	character	<b>dot</b> a
	width		
	Attribute		
	with fixed	Two dots above	
	character	a character	<b>ddot</b> a
	width		
	Attribute		
	with fixed	Three dots above a character	<b>dddot</b> a
	character		
*	width	Accent to the left above a grav character	
	Attribute		
	with fixed		<b>grave</b> a
	character		
	width	5 6 3	
,,	Attribute	Roof above a	<b>hat</b> a



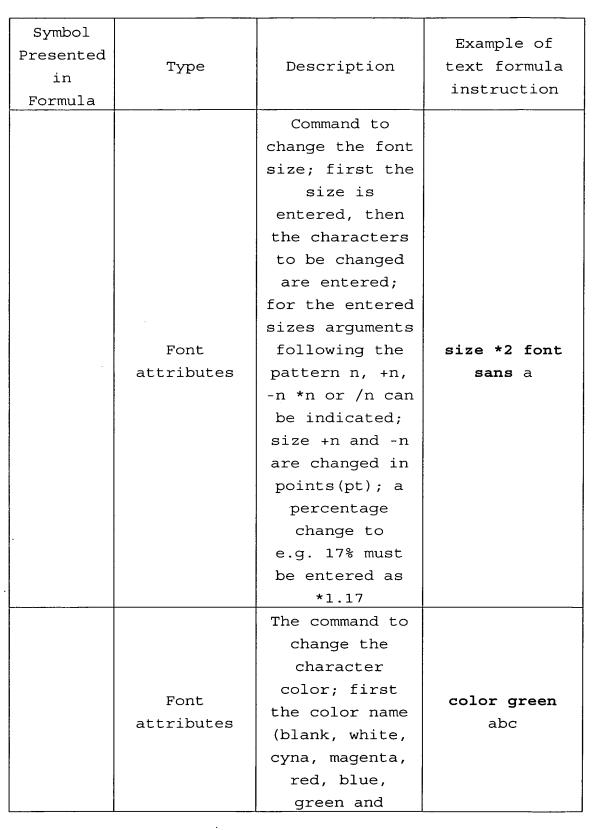


Symbol Presented in Formula	Туре	Description	Example of text formula instruction
	with fixed character width	character	
~	Attribute with fixed character width	Tilde above a character	<b>tilde</b> a
<b>→</b>	Attribute with fixed character width	Vector arrow above a character	<b>vec</b> a
<u> </u>	Attribute with variable character width	Horizontal bar below a character	<b>underline</b> a
ā	Attribute with variable character width	Horizontal bar above a character	<b>overline</b> a
<b>Q</b>	Attribute with variable character width	Horizontal bar through a character	<b>overstrike</b> a
<b>→</b>	Attribute with variable character width	Wide vector arrow, adjusts to the character size	<b>widevec</b> a





		-core	
Symbol ,			Example of
Presented	Туре	Description	text formula
in	- 11	_	instruction
Formula			111001000001
	Attribute		
	with	Wide tilde,	
~	variable	adjusts to the	widetilde
	character	character size	
	width		
	Attribute		
	with	Wide roof,	
^	variable	adjusts to the	widehat
	character	character size	
	width		
	Font		ital
	attributes	Italics	
	Font		nitalic
	attributes	Remove italics	
	Font	:	
	attributes	Bold	bold
	Font		
	attributes	Remove bold	nbold
	Font	Phantom	_
	attributes	character	phantom
		Command to	
		change	
	Font attributes	characters;	
		first the font	
		name (sans,	
		serif, or	<b>font sans</b> a
		fixed) is	
		entered, then	
		the characters	
		to be changed	
		are entered.	
	<del> </del>	are entered.	

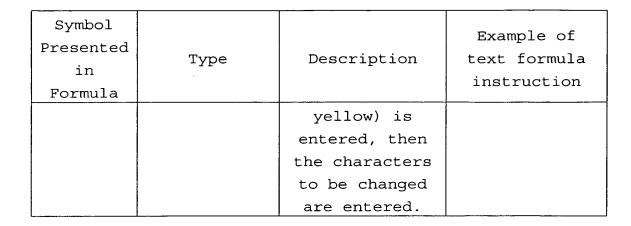


10

15

20

25



In addition to easy generation of a formula, the present invention includes an easy way to edit a data object like a mathematical formula. The object is entered by, e.g., a mouse click, on the object, and then is reconverted into the text formula instruction containing the text instruction symbols. The user edits the object by editing the text formula instruction, selects the edited text formula instruction again, and converts the same again into a data object, as described above. The editing operation can thus be carried out easily without entering a special tool like a formula editor.

Further, those of skill in the art will appreciate that while memory 311C is illustrated as one unit that can include both volatile memory and non-volatile memory, in most computer systems, memory 311C is implemented as a plurality of memory units. In more general terms, method 205 is stored in a computer readable medium, and when method 205 is loaded from the computer readable medium into a memory of a device, the device is configured to be a special purpose machine that executes method 205. Alternatively, the application used to execute method 220, e.g., application 319, may be stored in one computer readable

15

20

25

30

35



9

medium, and method 230 stored in another computer readable medium.

Also, herein, a computer program product comprises a medium configured to store or transport computer readable code for method 205, method 220, and/or method 230 or in which computer readable code for method 205, method 220, and/or method 230 is stored. Some examples of computer program products are CD-ROM discs, ROM cards, floppy discs, magnetic tapes, computer hard drives, servers on a network and signals transmitted over a network representing computer readable program code.

As illustrated in Figure 3A, this storage medium may belong to computer system 300C itself. However, the storage medium also may be removed from computer system 300C. For example, method 205 may be stored in either memory 311A or 311B that is physically located in a location different from processor 312C. The only requirement is that processor 312C is coupled to memory. This could be accomplished in a client-server system, e.g. system 300C is the client and system 300B is the server, or alternatively via a connection to another computer via modems and analog lines, or digital interfaces and a digital carrier line.

For example, memory 311C could be in a World Wide Web portal, while the display unit and processor are in a personal digital assistant (PDA), or a wireless telephone, for example, system 300A. Conversely, the display unit and at least one of the input devices could be in a client computer, a wireless telephone, or a PDA, while the memory and processor are part of a server computer on a wide area network, a local area network, or the Internet. In this paragraph, method 205 that includes the application used to perform method 220, as well as method 230 was

10

15

20

25

30

35



considered. However, those of skill in the art will appreciate that a similar description can be made for only method 220 and for only method 230. Accordingly, this description and that which follows is not repeated for each of the possible combinations and permutations for using and storing methods 220 and 230.

More specifically, computer system 300C, in one embodiment, can be a portable computer, a workstation, a two-way pager, a cellular telephone, a digital wireless telephone, a personal digital assistant, a server computer, an Internet appliance, or any other device that includes the components shown and that can execute method 205. Similarly, in another embodiment, computer system 300C can be comprised of multiple different computers, wireless devices, cellular telephones, digital telephones, two-way pagers, or personal digital assistants, server computers, or any desired combination of these devices that are interconnected to perform, method 205 as described herein. See, for example, Figure 3A.

Accordingly, a computer memory refers to a volatile memory, a non-volatile memory, or a combination of the two in any one of these devices. Similarly, a computer input unit and a display unit refers to the features providing the required functionality to input the information described herein, and to display the information described herein, respectively, in any one of the aforementioned or equivalent devices.

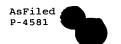
In view of this disclosure, method 230 and method 220 can be implemented in a wide variety of computer system configurations. In addition, method 205 could be stored as different modules in memories of different devices. For example, method 205 could initially be stored in a server computer, and

10

15

20

25



then as necessary, a module of method 205 could be transferred to a client device and executed on the client device. Consequently, part of method 205 would be executed on the server processor, and another part of method 205 would be executed on the client device. In view of this disclosure, those of skill in the art can implement the invention of a wide-variety of physical hardware configurations using an operating system and computer programming language of interest to the user.

In yet another embodiment illustrated in Figure 3B, method 205 is stored in memory 311B of system 300B. Stored method 205 is transferred, over network 315 to memory 311C in system 300C. In this embodiment, network interfaces 330B and 330C can be analog modems, digital modems, or a network interface card. If modems are used, network 315 includes a communications network, and method 205 is downloaded via the communications network.

While the invention has been particularly shown with reference to a preferred embodiment thereof, it will be understood by those skilled in the art that various other changes in the form and details may be made therein without departing from the spirit and scope of the invention.